

Living Atlas 1 Sprint - 2 Report

Long, Joshua
Kolb, Mitchell, William
Svetlik, Sierra Amelia

Introduction

This sprint started on February 20th and it ended on March 24th. This sprint focused on splitting up the work of the code between the three of us and to begin writing code for the project.

Retrospection

Long, Joshua

During sprint 1, the main focus of our team was to attend meetings with the client and carefully listen to their requirements and feedback. As a team member, my responsibility was to attend all the meetings and take detailed notes to ensure that we had a clear understanding of the client's vision for the project. Furthermore, I created a project overview and outlined the functional requirements for the project, which served as a helpful reference for the team for sprint 2.

Kolb, Mitchell, William

Looking back at last sprint I was tasked with "Have the clients give our team a starting data set of data to work off of in the database" and "Decide on a set amount of information that will be stored.". Both of these tasks were completed through the meetings that we have each week with the clients. We have streamlined the process of sharing documents by all joining a microsoft teams group. Because of this we have received demo datasets to work with and more details about what data the clients would like to store and where to get it from. We have been given some ".gov" websites that provide watershed data which will be useful in the future once we have the ability to import them.

Svetlik, Sierra Amelia

During the first sprint, I was assigned the task of deciding on the API we would use. This task was ultimately reassigned to Mitchell, but we did decide what API to use. We decided on RestAPI.

Result of Sprint - 2 Planning

Long, Joshua

I have shortlisted two key tasks that I believe will be essential to completing the project on time and to the satisfaction of our stakeholders.

The first task is to enable data fetching from the frontend. Currently, our application relies on a static dataset that is hard-coded into the frontend. By enabling data fetching, we will be able to dynamically retrieve data from the backend and display it in real-time on the frontend.

The second task is to enable connectivity between the frontend and backend. Currently, our application has been developed in isolation, with the frontend and backend running on separate servers. By enabling connectivity, we will be able to fully integrate the two components of our application.

The reason why I have chosen to prioritize these tasks at this point in the development process is because they represent critical components of our full-stack application that must be functioning properly in order for us to meet our project goals and deadlines.

Kolb, Mitchell, William

For Sprint 2 in regards to the backend we have 3 tasks that we wanted to complete by the end. The first task was to determine a common language scheme for the front end application to request information from the backend/database using the API endpoints. The second task was to have connected our database to the backend to query some type of data. The third task was to get code demonstrations of the four types of requests in API's. These would be a proof of concept for GET, POST, PUT, DELETE. We thought these tasks would be a good way to get the backend started as we also get the frontend and database configured as well. By completing these tasks we begin the process of opening the channels so that the frontend can retrieve data from the database.

Svetlik, Sierra Amelia

There are a couple tasks that need to be completed by the end of this sprint: connecting the database to the API. The front end and database are pretty disconnected throughout this project, the API being what will allow both to communicate, so connecting the database to the API will be an important step. Deciding on what information to store and how. While an example of a table to store users has been added, a table to store the data the app is being designed to store and access has not been because I don't know exactly what information should be stored along with the data, as well as how to access that data. Maybe talking to the client might prove helpful, if the customer has any idea how the data will be stored and accessed.

Sprint Task Assignment

Long, Joshua

I will be working on enabling data fetching from the frontend and enabling connectivity between the frontend and backend.

Kolb, Mitchell, William

I was tasked with starting the codebase for the backend. I have been using the REST API design principles to implement the http requests for GET, POST, PUT, DELETE. Along with that I have been tasked with getting a language scheme set up between the frontend and backend and trying to use the backend to connect the database once it is deployed using a third party service for testing.

Svetlik, Sierra Amelia

I will be creating the rest of the tables for the database, and helping Mitchell connect the database to the rest of the application.

Sprint Task(s) Details**Long, Joshua**

As the front-end developer, I was responsible for creating the basic layout of the application using React and I was successful in doing this. In addition to creating the basic layout of the application, I also designed and implemented a custom card component to display the data. This component was built with React and styled with CSS, and it allows users to easily view and interact with the information presented in the application.

Kolb, Mitchell, William

As the backend developer one of my tasks was "Use the REST API design principles to implement the http requests for GET, POST, PUT, DELETE. " Using Fast API and Python I would write code that is able to take a http request from the frontend to then be able to retrieve data from the database (GET), create new data (POST), update existing data in the database (PUT), and remove some data from the database (DELETE). If completed I should be able to use a program called insomnia to visually see the api work.

Svetlik, Sierra Amelia

Since my role involves creating the database for all the information and data storage, part of my responsibility was to decide on what information would be stored, and to try and get some examples of data to be stored in the database. We are using ElephantSQL to remotely store information, PostgreSQL as the database management system, and pgAdmin4 as the application. There are many different types of information that will be stored, including geographical information, as well as filetypes, such as .gdb.